UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/092,670	03/06/2002	Donald C. Soltis JR.	10016628-1	3827

7590 05/05/2006

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
MEONSKE, TONIA L	

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 05/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/092,670	SOLTIS ET AL.	
	Examiner	Art Unit	
	Tonia L. Meonske	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 January 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 October 2005 and 06 March 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

FRITZ FLEMING
Supervisory PRIMARY EXAMINER 5/1/2006
GROUP 2100
AU 2181

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. The drawings were received on October 26, 2005. These drawings are acceptable.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-19 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by Keckler, Stephen W., et al., Concurrent Event Handling through Multithreading, 1999, IEEE Transactions on Computers, volume 48, NO. 9, pages 903-916 (hereinafter "Keckler et al."). The rejections as set forth in the last office action mailed on July 26, 2005 are respectfully maintained and included below.

4. Referring to claim 1, Keckler et al. have taught a method for determining thread switch points within pipeline execution units of a processor, comprising the steps of:

a. monitoring instruction processing of a first thread within the pipeline execution units (page 908, left hand column, Monitors availability of instruction register operands.);

b. in the event of a possible switch point within the pipeline execution units, deactivating the first thread, or not, based upon a first urgency indicator for the first thread (page 908, left hand column, Each cycle the synchronization stage determines which instructions from each thread to activate and execute based on arrived data and associated instruction priorities.), the first urgency indicator being based upon progress of

the first thread within the pipeline execution units (page 908, left hand column, The first urgency indicator, or the indicator that indicates whether all of the data for the instruction has arrived, is based upon the progress of the thread within the pipeline execution units. When all of the data for an instruction is not available, the thread is not able to progress past the synchronization stage in the pipeline.).

5. Referring to claim 2, Keckler et al. have taught a method of claim 1, as described above, and further comprising deactivating the first thread and activating a second thread based upon a second urgency indicator for the second thread (page 908, left hand column, An important higher priority thread is now ready to execute and monopolizes the system.), the second urgency indicator being based upon expected progress of the second thread with the pipeline execution units (page 908, left hand column, A high priority thread is expected to progress through the pipeline execution units quicker than lower priority threads.).

6. Referring to claim 3, Keckler et al. have taught a method of claim 2, as described above, and further comprising deactivating the second thread, or not, based upon the second urgency indicator for the second thread and in the event of a possible switch point event of the second thread (page 908, left hand column, Each cycle instructions are scheduled for each thread depending on the respective priorities.).

7. Referring to claim 4, Keckler et al. have taught a method of claim 3, as described above, and further comprising activating another thread within the pipeline if the second thread is switched out (page 908, left hand column, If the input data for a high priority instruction is not available, another ready instruction from a different thread is activated and executed.).

8. Referring to claim 5, Keckler et al. have taught a method of claim 1, as described above, and the step of deactivating the first thread comprising deactivating the first thread, or not, based upon the first urgency indicator and upon a second urgency indicator of a second thread (page 908, left hand column, An important higher priority thread that is now ready to execute monopolizes the system.), the second urgency indicator being based upon expected progress of the second thread within the pipeline execution units (page 908, left hand column, A high priority thread is expected to progress through the pipeline execution units quicker than lower priority threads.).

9. Referring to claim 6, Keckler et al. have taught a method of claim 1, as described above, and the step of monitoring comprising utilizing a thread controller coupled with the execution units (page 908, left hand column, Synchronization stage).

10. Referring to claim 7, Keckler et al. have taught a method of claim 1, as described above, and further comprising modifying the first urgency indicator to increase or alternatively decrease urgency of the first thread based upon characteristics associated with the switch event (page 908, left hand column, An instruction can only stall for a maximum of 255 cycles and then is given a higher priority and is allowed to execute.).

11. Referring to claim 8, Keckler et al. have taught a method of claim 7, as described above, and further comprising determining whether a time slice expiration occurred (page 908, left hand column, The time slice is 255 cycles.).

12. Referring to claim 9, Keckler et al. have taught a method of claim 8, as described above, and further comprising utilizing a time slice expiration unit (page 908, left hand column, See preempt state, idle cycle counters, and limit registers.).

13. Referring to claim 10, Keckler et al. have taught a method of claim 7, as described above, and further comprising determining whether a cache miss occurred (Page 909, right hand column).

14. Referring to claim 11, Keckler et al. have taught a method of claim 7, as described above, and further comprising inserting an instruction to the pipeline to change urgency of the thread (page 908, Changing the urgency of the thread is inherently performed by an instruction.).

15. Referring to claim 12, Keckler et al. have taught a method of claim 1, as described above, and further comprising the steps of deactivating the first thread and activating a second thread, and modifying urgency of the second thread (page 908, left hand column, As soon as a high priority thread is ready to execute, the high priority instruction increases the urgency of the thread and monopolizes the system.).

16. Referring to claim 13, Keckler et al. have taught a method of claim 1, as described above, and further comprising the steps of monitoring possible switch points of an inactive thread having a second urgency indicator that is based upon expected progress of the inactive thread within the pipeline execution units (page 908, left hand column, A second urgency indicator is the instruction thread priority indication of Keckler. A high priority thread is expected to progress through the pipeline execution units quicker than lower priority threads.), and deactivating the first thread, or not, based upon a first and second urgency indicator (page 908, left hand column, The time slice is 255 cycles, This is performed on a cycle-by-cycle basis.).

17. Referring to claim 14, Keckler et al. have taught a processor for processing multi-threaded program instructions, comprising:

a. an array of pipeline execution units and associated heuristics affecting how the instructions are processed within the units (page 904, left hand column, page 908); and

b. a thread controller for monitoring processing of the instructions within the units and for switching between multiple program threads based upon (a) the heuristics and (b) urgencies of the program threads (page 908, left hand column, Switch between multiple program threads based on (a) the time slice duration of 255 cycles, or the heuristics, and (b) the thread priorities, or the urgencies of the program threads.),

wherein the urgencies are based upon one or both of (a) progress of the threads through the pipeline execution units (page 908, left hand column, The first urgency indicator, or the indicator that indicates whether all of the data for the instruction has arrived, is based upon the progress of the thread within the pipeline execution units. When all of the data for an instruction is not available, the thread is not able to progress past the synchronization stage in the pipeline.) and (b) expected progress of the program threads through the pipeline execution units (page 908, left hand column, A high priority thread is expected to progress through the pipeline execution units quicker than lower priority threads.).

18. Referring to claim 15, Keckler et al. have taught a processor of claim 14, as described above, and the heuristics comprising one or more of time slice expiration heuristics, cache miss heuristics and processor interrupt heuristics (page 908, left hand column, The time slice is 255 cycles.).

19. Referring to claim 16, Keckler et al. have taught a processor of claim 14, as described above, and the program threads comprising one or more instructions, one of the instructions

changing urgency for at least one thread of the processor (page 908, left hand column, A higher priority instruction becomes ready to execute and the urgency of the instruction increases, thereby effectively decreasing urgency of the other threads.).

20. Referring to claim 17, Keckler et al. have taught a processor of claim 14, as described above, and the controller modifying an urgency of any of the threads to modify future treatment of the threads in switch out events (page 908, left hand column, When a thread has been idle for 255 cycles, the urgency of the instruction is modified in order to allow the thread to execute.).

21. Referring to claim 18, Keckler et al. have taught a processor of claim 17, as described above, and the controller either decreasing or increasing urgency for the program threads by injecting an instruction to the pipeline execution units (page 908, Changing the urgency of the thread is inherently performed by an instruction.).

22. Referring to claim 19, Keckler et al. have taught a processor of claim 12, as described above, and further comprising a time slice expiration unit for monitoring expiration of threads within the processor (page 908, left hand column, The time slice is 255 cycles. See preempt state, idle cycle counters, and limit registers.).

Response to Arguments

23. Applicant's arguments filed January 31, 2006 have been fully considered but they are not persuasive.

24. On pages 3 and 4, Applicant argues with respect to claim 1 in essence:

> *"Keckler does not disclose or suggest an urgency indicator upon which deactivation of a thread is based. ... Keckler shows no relationship between an instruction's operand availability and progress of a thread through a pipeline. Rather, Keckler discloses that "if an instructions input data is not available, the instruction will not execute. Thus the unavailability of an instructions operands causes the pipeline to stall, and is clearly not an indicator of a thread's progress through a pipeline.""*

However, the operand indicators of Keckler do indicate the progress of a thread through a pipeline. Executing an instruction thread requires many different steps in each stage of the pipeline. Retrieving data is a part of the overall instruction execution in the pipeline, specifically it is a step in the synchronization stage. Each time data is retrieved, the instruction has made progress and the associated operand validity indicator is updated as such. Therefore the urgency indicators of Keckler are in fact indicators of a threads progress through the pipeline (The operand validity bits indicate how far the thread has progressed in the synchronization stage of the pipeline.). Therefore this argument is moot.

25. On page 4, Applicant argues with respect to claim 2 in essence:

"Keckler does not disclose or suggest a second urgency indicator of a second thread to decide upon deactivating the first thread and activating a second thread, as required by claim 2."

However, Keckler has taught "deactivating the first thread and activating a second thread based upon a second urgency indicator for the second thread" (page 908, left had column), as in claim 2. The claimed urgency indicators include the valid bits for instruction operands, the priority of the thread, and the limit register value of Keckler. In Keckler when a first thread is monopolizing the system and an important higher priority thread becomes ready to execute by changing one of it's urgency indicators, the higher priority thread monopolizes the system. In this case the first thread is deactivated and the important higher priority thread is activated, based upon a second urgency indicator. Therefore this argument is moot.

26. On page 4, Applicant argues with respect to claim 3 in essence:

"Keckler does not disclose deactivating the second thread or not, based upon the second urgency indicator for the second thread and in the event of a possible switch point event of the second thread, as required by claim 3."

However, Keckler has taught deactivating the second thread, or not, based upon the second urgency indicator for the second thread and in the event of a possible switch point event of the second thread (page 908, left hand column). The claimed urgency indicators include the valid bits for instruction operands, the priority of the thread, and the limit register value of Keckler. Each cycle a switch point event can possibly occur. Each cycle instructions are scheduled for each thread depending on their urgency indicators. Each cycle, or in the event of a possible switch point event, the synchronization stage determines whether to schedule the instruction thread or not, based upon the urgency indicator for the second instruction. Therefore this argument is moot.

27. On page 4, Applicant argues with respect to claim 4 in essence:

"Keckler does not disclose or suggest activating another thread within the pipeline if the second thread is switched out, as in claim 4."

However, Keckler has taught activating another thread within the pipeline if the second thread is switched out (page 908, left hand column). When the input data for a high priority instruction of Keckler is not available, another ready instruction from a different thread is activated and executed. Therefore this argument is moot.

28. On page 4, Applicant argues with respect to claim 5 in essence:

"Keckler does not disclose an urgency indicator being based upon expected progress of a thread."

However, Keckler has taught an urgency indicator being based upon expected progress of a thread (page 908, left hand column). The claimed urgency indicators include the valid bits for instruction operands, the priority of the thread, and the limit register value of Keckler. A high priority thread is expected to progress through the pipeline execution units quicker than lower priority threads.).

29. On pages 4 and 5, Applicant argues with respect to claim 6 in essence:

"Keckler's synchronization stage operates upon instructions, and does not monitor instruction processing of a first thread."

However, Keckler has taught monitoring instruction processing of a first thread (page 908, left hand column). Keckler has taught that processing an instruction thread requires many different steps in each stage of the pipeline. Retrieving data is a part of the overall instruction processing in the pipeline, specifically it is a step in the synchronization stage. Each time data is retrieved, the instruction has made progress and the associated operand validity indicator is updated as such. The synchronization stage monitors the instruction processing of a thread, specifically the stage monitors the data processed, or retrieved, for an instruction. The synchronization stage monitors the reservation station for the highest priority instruction thread with all operands available and executes that thread. Therefore this argument is moot.

30. On page 5, Applicant argues with respect to claim 7 in essence:

"Keckler does not teach or suggest modifying an urgency indicator based upon associated possible switch point."

However, claim 7 states modifying the first urgency indicator to increase or alternatively

decrease urgency of the first thread based upon characteristics associated with the possible switch point. In Keckler each cycle there is a possible switch point. An instruction can only stall for a maximum of 255 cycles and then is given a higher priority and is allowed to execute. Keckler has taught that the urgency indicator of a thread (idle cycle counter) is increased each cycle that it does not issue, such that the thread becomes more urgent (page 908, left hand column). So Keckler has in fact taught modifying an urgency indicator based upon an associated possible switch point. Therefore this argument is moot.

31. On page 5, Applicant argues with respect to claims 8, 9, and 15 in essence:

"As known in the art, time slicing provides a method of ensuring equal periods of processor utilization between threads. We contend that the preempt state of Keckler is not equivalent, at least because this preempt state does not divide processor utilization equally between threads, and is only used when "one thread is creating a tremendous number of events," such that "other threads will be able to continue making forward progress.""

However, Applicant is arguing a feature of the invention not specifically stated in the claim language, which is improper. Claimed subject matter, not the specification, is the measure of invention. Limitations in the specification cannot be read into the claims for the purpose of avoiding the prior art. In re Self, 213 USPQ 1,5 (CCPA 1982); In re Priest, 199 USPQ 11,15 (CCPA 1978).

> *"It is the claims that measure the invention." SRI Int'l v. Matshshita Elec. Corp., 775 F.2d 1107, 1121, 227 USPQ 577, 585 (Fed. Cir. 1985) (en banc).*
>
> *"The invention disclosed in Hiniker's written description may be outstanding in its field, but the name of the game is the claim." In re Hiniker Co., 47 USPQ2d 1523, 1529 (Fed. Cir. 1998).*
>
> *"[A]s an initial matter, the PTO applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they*

> *would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification." In re Morris, 44 USPQ2d 1023, 1027 (Fed. Cir. 1997).*
>
> *"limitations appearing in the specification will not be read into the claims, and ... interpreting what is meant by a word in a claim 'is not to be confused with adding an extraneous limitation appearing in the specification, which is improper'." Intervet Am., v. Kee-Vet Labs., 12 USPQ2d 1474, 1476 (Fed. Cir. 1989)(citation omitted).*
>
> *"it is entirely proper to use the specification to interpret what the patentee meant by a word or phrase in the claim, ... this is not to be confused with adding an extraneous limitation appearing in the specification, which is improper. By 'extraneous,' we mean a limitation read into a claim from the specification wholly apart from any need to interpret ... particular words or phrases in the claim." In re Paulsen, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994) (citation omitted).*

In this case applicant has not specifically claimed that time slicing provides a method of ensuring equal periods of processor utilization between threads. If applicant would like a specific definition of time slicing read into the claims, then applicant should specifically claim that definition. Therefore this argument is moot.

32. On page 5, Applicant argues with respect to claim 10 in essence:

"Claim 10 recites determining whether a cache miss occurred. ... Keckler has no need to detect cache misses, since operation of the MAP chip does not depend upon them."

However, in order for the external memory interface hardware to format an event record when a TLB miss occurs, or a cache miss occurs, Keckler has to detect cache misses. Otherwise the event record would not get formatted when a cache miss occurs. Therefore this argument is moot.

33. On page 6, Applicant argues with respect to claim 11, and similarly with respect to claim 12 in essence:

"Keckler has no teaching or suggestion of modifying a thread's urgency indicator by

inserting an instruction to the pipeline, as required by claim 11."

However, Keckler has taught inserting an instruction to the pipeline to change the urgency of the thread (page 908). In order for the urgencies to change in value, they must be instructed to do so. Changing the urgency of the thread is necessarily performed by an instruction. Therefore this argument is moot.

34. On page 6, Applicant argues with respect to claim 13 in essence:

"As argued above, the priority system of Keckler is not equivalent to the urgency indicator of the immediate application. The priority system of Keckler is static and is not based upon expected progress of a thread. The priority system of Keckler provides a selection decision between threads based upon a desired progress and not an expected progress."

However, threads that are desired to progress quickly through the pipeline are assigned a high priority value. Threads with a higher priority are expected to progress through the pipeline quicker than lower priority threads. Therefore this argument is moot.

35. On pages 6 and 7, Applicant argues with respect to claim 14 in essence:

"Keckler does not disclose or suggest a thread controller for monitoring processing of the instructions within pipeline execution units and for switching between multiple program threads based upon heuristics and urgencies of the program threads, as required by element ii of claim 14."

However, Keckler has taught a thread controller for monitoring processing of the instructions within the units and for switching between multiple program threads based upon (a) the heuristics and (b) urgencies of the program threads (page 908, left hand column). Keckler switches between multiple program threads based on (a) the time slice duration of 255 cycles, or the heuristics, and (b) the thread priorities, or the urgencies of

the program threads.). Therefore this argument is moot

36. On page 7, Applicant argues with respect to claim 16 in essence:

"Keckler does not disclose or suggest program threads with one of the instructions changing urgency of at least one thread of the processor, as in claim 16."
However, Keckler has taught one of the instructions changing urgency for at least one thread of the processor (page 908, left hand column). In Keckler a higher priority instruction becomes ready to execute and the urgency of the instruction increases, thereby effectively decreasing urgency of the other threads. Therefore this argument is moot.

37. On page 8, Applicant argues with respect to claim 17 in essence:

"Keckler does not disclose or describe a controller modifying an urgency of any of the threads to modify future treatment of the threads of switch out events. And Keckler does not disclose that priority of a thread is modified, nor urgency of a thread."

However, Keckler has taught a controller modifying an urgency of any of the threads to modify future treatment of the threads in switch out events (page 908, left hand column). Each cycle a thread des not execute, it becomes more urgent by increasing an idle cycle counter register. When a thread has been idle for 255 cycles, the urgency of the instruction is modified in order to allow the thread to execute. Therefore this argument is moot.

38. On page 8, Applicant argues with respect to claim 18 in essence:

"Nowhere within Keckler is there disclosure of a controller decreasing or increasing the urgency (or priority) for the program threads by injecting an instruction into the pipeline execution units, as required by claim 18."

However, Keckler has taught a controller either decreasing or increasing urgency for the program threads by injecting an instruction to the pipeline execution units (page 908). In order for the urgencies (priorities) to change in value, they must be instructed to do so.

Changing the urgency of the thread is necessarily performed by an instruction. Therefore this argument is moot.

39. On page 8, Applicant argues with respect to claim 19 in essence:

"Keckler does not disclose or suggest a time slice expiration unit for monitoring expiration of threads with the processor, as required by claim 19, since, as argued above, Keckler des not operate with time slicing."

However, Keckler does operate with time slicing to the extent claimed (page 908, left hand column). The time slice is 255 cycles. See preempt state, idle cycle counters, and limit registers. A time slice expiration unit monitors for when a thread waits 255 cycles, upon which a thread expires and issues. Therefore Keckler has in fact taught a time slice expiration unit for monitoring expiration of threads with the processor, as required by claim 19. Therefore this argument is moot.

Conclusion

40. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

41. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

42. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tonia L. Meonske whose telephone number is (571) 272-4170. The examiner can normally be reached on Monday-Friday, with every other Friday off.

43. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

44. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

tlm

Ftzm. Fleming

FRITZ FLEMING
PRIMARY EXAMINER
GROUP 2100

Supervisory 5/1/2006

AU 2181